Detailed Radiation Fault Modeling of the Remote Exploration and Experimentation (REE) First Generation Testbed Architecture

John Beahan
Jet Propulsion Laboratory
4800 Oak Grove Dr
Pasadena, CA 91109
818-356-0469
John.Beahan@jpl.nasa.gov

Larry Edmonds
Jet Propulsion Laboratory
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-2778
Larry.D.Edmonds@jpl.nasa.gov

Robert D. Ferraro
Jet Propulsion Laboratory
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-1340
Robert.D.Ferraro@jpl.nasa.gov

Allan Johnston
Jet Propulsion Laboratory
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-6425
Allan.H.Johnston@jpl.nasa.gov

Raphael R. Some
Jet Propulsion Laboratory
4800 Oak Grove Dr
Pasadena, CA 91109
818-354-3055
Rsome@jpl.nasa.gov

Abstract--The goal of the NASA HPCC Remote Exploration and Experimentation (REE) Project is to transfer commercial supercomputing technology into space. The project will use state of the art, low-power, non-radiation-hardened, Commercial Off-The-Shelf (COTS) hardware chips and COTS software to the maximum extent possible, and will rely on Software-Implemented Fault Tolerance (SIFT) to provide the required levels of availability and reliability. In this paper, we outline the methodology used to develop a detailed radiation fault model for the REE Testbed architecture, and plans for validation of this model. The model addresses the effects of energetic protons and heavy ions which cause Single Event Upset (SEU) and Single Event Multiple Upset (SEMU) events in digital logic devices and which are expected to be the primary fault generation mechanism. Unlike previous modeling efforts, this model will address fault rates and types in computer subsystems at a sufficiently fine level of granularity (i.e., the register level) that specific software and operational errors can be derived. We present the current state of the model, model verification activities and results to date, and plans for the future. Finally, we explain the methodology by which this model will be used to drive a higher level tool that generates application-level error effects sets. These error effects sets will be used in conjunction with our Testbed fault injection capabilities and our applications' mission scenarios to replicate the predicted fault environment on our suite of onboard applications.

1. Introduction:

NASA's future spaceborne science missions are evolving in directions that will require substantial onboard computing capabilities for both near earth and deep space exploration. Downlink bandwidth limitations and excessive round trip communication delays are motivating the increased use of onboard computing to enhance the science value of missions and, in some cases, to enable the missions themselves. Projects such as the Gamma Ray Large Area Space Telescope (GLAST), the Next Generation Space Telescope (NGST) and autonomous rovers being designed for Mars exploration in the next millennium already require some onboard computing capabilities to either enable or to greatly enhance their baseline missions. The difficulty NASA is encountering is that radiation hardened components are both extremely expensive and lag several generations behind the commercial state of the art. The Remote Exploration and Experimentation (REE) Project is working to mitigate this problem by migrating ground based commercial scalable computing technologies into space in a timely and cost effective manner. The approach being taken is to exploit a comprehensive architectural strategy that incorporates a custom, but architecturally insensitive, Software Implemented Fault Tolerance (SIFT) middleware layer, as well as a generic library of Algorithm-Based Fault Tolerance (ABFT) techniques, to enable the direct use of latest generation commercial hardware and software components in future space systems. This strategy will allow high throughput computation even in the presence of relatively high rates of radiation induced transient upsets as well as in the presence of permanent faults. A First Generation Testbed, equipped with fault injection capabilities, is being constructed out of COTS hardware and software to test these concepts.

Unlike the development of a system composed of radiation hardened components, in which the baseline technology and the circuitry are designed to be insensitive to the worst case expected radiation environment, the development of an efficient SIFT based system requires a high fidelity, realistic radiation effects model. Further, SIFT development requires an accurate and validated fault to error translation model as well as an error propagation model. The reasons for these requirements are due to the nature of a SIFT system in which the fault (i.e., the physical, hardware-level effect) is not prevented as it is in a radiation hardened computer, but rather is allowed to occur. The error (the logical manifestation of the fault as seen by the system or software) or a subsequent manifestation due to propagation of the error is then detected and handled by the SIFT software. Clearly, the design of such a system is dependent on a detailed understanding of the types of faults which will occur, the errors generated by these faults and the rate at which they will appear. Armed with this information, an efficient SIFT system may be designed. Here, efficient refers to the notion that over-design of the system is wasteful in power/performance and, in a spacecraft environment, is almost as bad as an under-designed system in which the faults are not detected and handled in a timely manner, resulting in poor throughput, reliability and availability.

In the case of the REE system, there are several additional characteristics which allow the design of a machine with exceptional power/performance ratio (for a space based computer), but which also require complete, thoroughly validated, high fidelity fault/error models.

1. The REE system is not intended for use in high radiation environments such as the Van Allen Belts or the Jovian System. This is key to the ability to use non-radiation hardened components, and thus gain a two to three generation advantage over available radiation hardened flight computers.

2. The REE system is being designed primarily for the processing of science data, rather than hard (vs. soft) real time, mission critical, spacecraft control functions. Thus, occasional resets, processing delays, and possibly even dropped frames or other service interruptions are acceptable. The advantage here is that we can use non-replicated fault tolerance techniques with concomitant advantages in power/peformance.

3. The system is intended, with appropriate replication techniques, such as software implemented triple-modular-redundancy, to be capable of performing a limited range of real time tasks. This will be, at least initially, in a segregated portion of the system which will operate in a relatively poor power/performance mode (providing only a 2.5 to 3.0 power/performance improvement over available radiation hardened computers vs the expected 10X improvement in the rest of the system). This segregation of real time activities will allow the system to perform these types of tasks if necessary, but with resultant penalties. In the future, we plan to investigate the possibility of performing real time tasks in a minimally or non-replicated and non-segregated mode.

The above require that we accurately predict what types of errors will be generated, under what conditions the system will become bogged down in error handling, and under what conditions errors will propagate through the system error detection and containment boundaries. Detailed error models will be crucial for the design of appropriate (efficient) error detection and handling techniques and an understanding of when and how to invoke them for a given environment and application software set.

The primary concern, for the REE environments, i.e., Low Earth Orbit (LEO), Geosynchronous Earth Orbit (GEO) and Deep Space, is transient errors induced by natural Galactic Cosmic Rays (GCR's), and energetic protons. The principle faults are single bit flips in memory and registers, though there is growing concern that transients in clock lines and non-clocked logic may occur with deep sub-micron feature sizes [3]. These latter effects may cause significantly worse faults as they can propagate to large numbers of bits/registers in an unpredictable and wide ranging manner. In addition to single bit flips, there is an expectation due to both analysis and physical evidence of an increasing rate of multiple bit flips due to shrinking feature sizes [3]. In this fault mode, several physically adjacent cells may be disrupted by the passage of a single energetic particle. Finally, while total dose radiation effects are not believed to be significant for REE missions in terms of component performance degradation, there is a concern that long-term exposure to stresses such as total dose radiation and thermal cycling may increase SEU susceptibility. Thus it is important that degradation from long term, low-level stresses be understood and accounted for.

It is apparent that the key to an appropriate fault tolerance strategy for the REE system will be an accurate understanding of the fault environment including fault rates/types and error propagation modes. Over estimation of fault rates results in overly conservative design and high power and throughput penalties. Underestimation results in unacceptable system availability and failure rates. Similarly, overlooking even a small number of error types and propagation modes could have serious deleterious effects in a fielded system. In order to derive adequate fault/error detection and handling strategies it is critical to have a high fidelity fault/error model. To our knowledge, relatively little work has been done at the required levels of detail, especially with state of the art, deep submicron COTS components and with complex modern architectures.

2. Previous and Related Work

As stated above, there has been relatively little work done in the regime of interest. The following is a brief discussion of the deficiencies in the state of the art:

1. Most experience to date has been with older, larger feature size devices where faults are localized to memory and registers [1, 2]. At some point, we expect

to see faults in non-clocked logic, but exactly where this begins to be manifested is unknown [3, 4]. As we proceed down the technology curve to 0.10 micron and smaller feature sizes, it is crucial that we understand where and how these types of faults begin to occur. It will be critical to develop techniques to rapidly detect and handle these types of faults, as they will, doubtless, cause severe (and currently unpredictable) processing disruption at the node and system levels.

2. Multiple bit errors have occurred infrequently and have not been a serious issue for most systems. To date, the use of standard single-error-correcting codes has been adequate for memory systems fabricated with unhardened components. As we proceed to the deep submicron level, multiple bit errors will likely be seen both in memory systems (including caches) and in registers. The overhead associated with multiple bit correcting codes is substantial and may, in some cases, be impractical (such as in on-chip caches). The design of a realistically deployable system (taking into account realistic power/mass/volume constraints of space based systems) will require detailed knowledge of both the probabilities of occurrence and the propagation characteristics of these fault types.

3. Previous investigations into radiation effects were for the purposes of worst case analysis and design of so called 'hard fault tolerant' systems, based, for the most part, on hardware implemented fault avoidance and/or tolerance. The resulting models have been extremely conservative and would, if used in a SIFT based system, result in over-design of the system and poor power/performance. It is necessary that realistic fault/error models be used for SIFT based designs.

4. Due to the rigors of a SIFT based design, a thorough and detailed understanding of the component and system architectures is required for proper execution of the design and validation tasks. This simply was not previously necessary and, for the most part, too expensive and difficult to justify. Further, the systems themselves, i.e., fielded space based computer systems, were not very complex. They tended to be single computer systems with relatively simple processor architectures. There have been no studies of any systems of the complexity of an REE type system, i.e., 20 or more state-of-the-art processors connected via a high speed switched packet network. Finally, while there have been some attempts to look into the behavior of distributed processors under fault conditions, we are unaware of any study of a system approaching the complexity of a parallel-processing supercomputer, or under the high fault-rate conditions seen by COTS in space environments.

The following are some anecdotal examples of recent studies and their outcomes:

1. Experimental results from the Japanese HITEN space probe, indicated a lower than expected level of faults in Low Earth Orbit (LEO), resulting in only a few dozen faults over a period of several months. The HITEN computer was an extremely simple processor and memory system of 1980's vintage. A fairly detailed model was generated for this system, yet even this simple system defied correct analysis (though to be fair, the designers were, once again, concerned with worst case behavior rather than expected or probable effects).

2. Some of the most applicable work to date was performed at Chalmers University, approximately 6 years ago [5,6]. In this series of studies, a radiation effects fault model was developed based on single-bit flips applied to a VHDL gate-level simulation of a RISC CPU. The single-bit-flip model was previously validated experimentally using a radioactive Californium source to irradiate a similar physical device. Detailed analysis of single-bit faults injected in to simulated devices was performed. Comparing the outcomes of radiation based fault injection and VHDL model simulated fault injection against what could be achieved using only software-implemented fault injection on unmodified computing hardware, it was determined that software methods were capable of emulating 98-99% of all single-bit fault effects for the candidate architecture. These results provide an indication that it is possible, without the addition of special fault-injection hardware, to simulate the effects of the large majority of bit flip faults in RISC architectures. While this work is arguably some of the most creative and original work done in the field, it was never validated in space based experiments. It is also not apparent that the Californium source is equivalent to naturally occurring space radiation in terms of energy deposited, angle of deposition, or SEU effect. And finally, the architectures being tested in this manner were simple compared with modern systems. Even so, the experiments remain, in our opinion, some of the best performed to date in attempting to obtain high fidelity models for the purpose of understanding error generation and propagation for fault tolerance analysis.

3. Perhaps one of the best recent examples of the radiation effects characterization of a commercial component were the recent heavy ion experiments performed on the Power PC 603e which characterized faults in the processor register set and internal caches [9]. While useful for worst case analysis and to obtain general SEU susceptibility data, there is insufficient detail to provide a clear understanding of what errors were seen in the various functional blocks of the device or how they might be manifest under actual operating conditions. This data was, however, useful in generating our 'first cut' models in that, when combined with additional information on the device architecture and layout, a reasonable set of estimations

could be made regarding probable fault modes and error effects.

3. Fault Effects Modeling and Design Methodology

The development of the fault-tolerant REE multicomputer will require several activities to be carried out as part of a unified fault modeling and system development methodology, including:

1. Building a low-level fault model for the hardware,
2. Verifying the model experimentally with ground and flight radiation tests,
3. Building system and software-level error generation/propagation models,
4. Verifying those models experimentally,
5. Performing fault injection experiments on both the hardware and software during development to guide the design process, and finally
6. Validating that the developed system will operate correctly under expected radiation conditions.

Because of the fact that many of the listed activities will take extended periods of time, many of the modeling and verification processes must be done in parallel with the system development (see Figure 1). We are therefore developing a set of initial fault and error models and a software-based fault injection facility. These will be used for experimentation to guide development, both to emulate space radiation conditions, and to perform sensitivity studies with specific types of faults injected, at accelerated rates, into specific hardware and software targets, under controlled computing-load conditions.

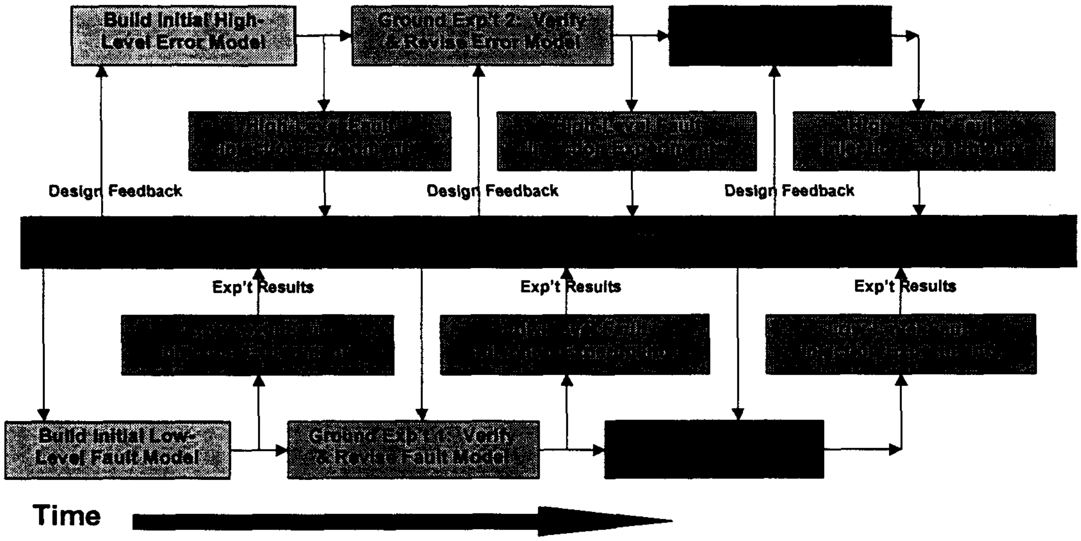
Build Initial High-Level Error Model
Ground Exp't 2: Verify & Revise Error Model
Design Feedback
Design Feedback
Design Feedback
Exp't Results
Exp't Results
Exp't Results
Build Initial Low-Level Fault Model
Ground Exp't 1: Verify & Revise Fault Model
Time

Figure 1: Fault Modeling and System Development Methodology

In general, the prediction of faults is difficult, requiring in-depth analysis of semiconductor and radiation physics. The conversion of the fault to an error and the analysis of error propagation is, by comparison, somewhat simpler. Our strategy is to emulate radiation faults in COTS CPUs, memories and other components. Previous software based fault injector developments and associated studies indicate that this is a reasonable and relatively straightforward approach and should be sufficient for early studies prior to physical irradiation experiments. The least studied and potentially most significant area of software fault emulation is that of caches (especially on-chip caches), and Memory Management Units. To our knowledge, there have as yet been no detailed modeling or experimentation regarding the effects of radiation on caches or virtual memory hardware, and these may be potentially problematic because of the ease with which single faults could propagate to cause multiple errors. Our initial fault modeling effort therefore concentrates on cache and MMU fault effects.

4. Modeling Assumptions:

The assumptions we are using for this fault modeling and system development methodology include:

Faults are primarily caused by Single-Event Upsets (SEUs) due to heavy ions from the Galactic Cosmic Ray background, and energetic protons from the solar wind or trapped radiation belts. Single-Event Multiple Upsets are relatively rare. For this phase of the study, we assume that Single-Event Latchup effects will be extremely rare, and non-destructive when they do occur, due to the prevalence of epitaxial layers in chips. They are therefor ignored at this time and will be considered in a subsequent model enhancement.

Faults are grouped into two types:

Latch faults, including latches, flip-flops, memory cells, and any other structure that persistently stores a bit. 'Visible' latches are included, such as data registers, as are 'invisible' latches, which are used to implement structures such as instruction pipeline stages and processor register reservation scoreboards.

Gate faults, which occur when a SEU event happens at approximately the same time period as a clock transition, causing the gate to flip its effective bit value.

Due to the tight timing required for a gate fault to propagate to, and be latched by, a register, faults in latches are 10's of thousands of times more likely than faults in gate level, or combinatorial logic. As clock rates increase, this difference will shrink, due to the increased fraction of time available for combinatorial logic to present erroneous values to registers which may then latch these transients. This is owing to the fact that, as clock rate rises, the number of latching windows increases faster than the width of each window decreases. [Clock line faults, in which a SEU induced transient on a clock line may cause one or more registers to latch at an incorrect time, possibly latching an incorrect value from the associated combinatorial logic feeding the register data input lines, are treated separately and discussed in a later section of this paper.] Future experimentation will validate and further quantify assumptions.

Single-bit errors in RAM and L2 cache are estimated, but will be corrected by EDAC, and so are not brought up to the system level. Multiple-bit faults are initially assumed to be 100,000 times less likely than single-bit faults. We believe this to be a relatively conservative estimate. An exact Poisson rate calculation will be performed, and validated at a later date.

Single Event Functional Interrupts (SEFI) faults, which are defined as any interruption of the function of a device such as an SEU-induced mode-change from standard operational mode to test mode in a DRAM, are assumed, for this initial modeling activity, to be only a small subset of errors caused by faults. Future physical experimentation will validate and further quantify this assumption.

5. Fault Model Structure

The methodology we are employing uses a multi-layered model for radiation effects, and is divided into two main sections (see Figure 2). The bottom section is composed of two layers, and models the initial occurrence of faults. The lowest layer is termed the *physical level* model, and calculates the effects of radiation at the level of individual latches and gates. It takes into account (a) the space radiation environment and (b) the geometry, feature size and material composition of the semiconductor technology. The second layer is called the *design level* model, and takes into account the number of gates and the number of bit storage elements in each chip, including latches, flip-flops, memory cells, etc. and derives an expected rate of faults that will be experienced by each device. For simplicity, in this document we refer to the different kinds of bit-storage elements using the generic term *latches.* The level of detail that is available about the number of latches and gates in each functional unit within a device will determine how closely the fault model can predict the location, frequency and effects of faults within the device.

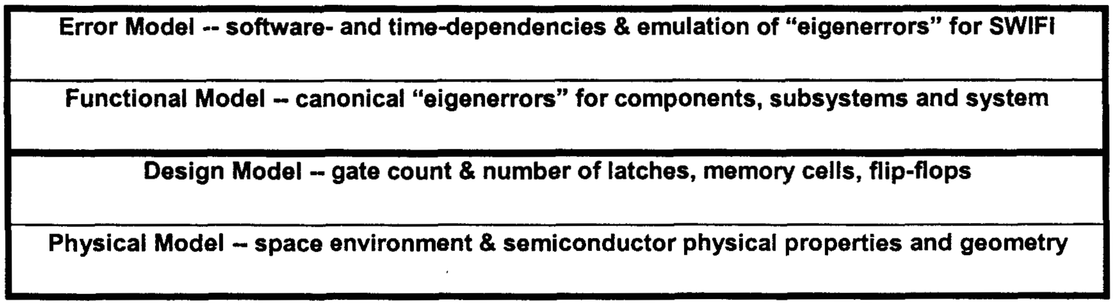
Error Model -- software- and time-dependencies & emulation of "eigenerrors" for SWIFI
Functional Model -- canonical "eigenerrors" for components, subsystems and system
Design Model -- gate count & number of latches, memory cells, flip-flops
Physical Model -- space environment & semiconductor physical properties and geometry

Figure 2: Layered Fault Model

Excepting the physical radiation effects characterization work being performed to enable the modeling activity, only software-based fault injection is being used for the REE development. This is partly because of the desire to use COTS board-level hardware systems as development platforms, and partly because we feel it is a safe and cost-effective approach. This software fault injection approach means that many classes of faults must be emulated at the software level, in particular those due to errors in caches and the MMUs, since no route exists for injecting faults into the caches, TLB's or other related modules. The upper half of our model is devoted to modeling the error effects on the software, and on the operation of the integrated hardware/software system, that will result from radiation faults. This level also addresses issues associated with effective software emulation of hardware faults.

The upper half of the multi-layered fault model is also composed of two layers, the lower of which is a *functional level* model of the system. It categorizes all the possible error behaviors of all its modules, devices, subsystems and itself as a whole into a few classes called *eigenerrors*. The uppermost layer of the fault model is the *error level*, in which parameterized models of the software and data being run by the system are integrated with the preceding hardware models to predict the rates of the different eigenerrors in the operational system. These rates can then be used as inputs to the fault injectors used to evaluate reliability and performance in the various system designs being analyzed.

A detailed description of the model follows. Note that initial versions of only the physical level and design level have been developed so far. Work is still in progress on the functional and error level models, as well as continuing refinement of the model's lower levels.

Physical Level (initial version completed)

1. A space environment factor is used to adjust fault rates to the desired mission conditions. This factor accounts for the distribution of particles and the radiation flux density present. Due to differences in the effects of protons vs. heavy ions, multiple space environment factors are used. Each chip has an environmental factor set to accommodate the effects of local shielding.

2. A technology factor is used to adjust for changes in fault rate properties due to changes in semiconductor materials and fabrication. Each chip has an individual parameter.

3. A per-latch fault rate is used to capture the device-level likelihood that a single latch (flip-flop) will experience a single bit-flip fault. Each type of chip has an individual parameter to account for multiple technologies being used in the system design.

4. A per-gate fault rate is used to capture the device-level likelihood that a single gate will experience a single bit-flip fault. Each type of chip has an individual parameter.

Design Level (initial version completed)

5. Each chip is modeled parametrically in terms of latches, and latch faults (single bit-flips) are estimated thereby.

6. Large chips are also modeled in terms of number of gates, either with specific gate counts or approximation using percentage of chip area, and gate faults (single bit-flips) are estimated thereby.

Functional Level (under construction)

7. The functional modules of each device are examined to determine the effects on their behavior resulting from faults. This includes both faults that occur internally, and faults that are fed in as inputs.

8. Each device is examined as a discrete functional entity to determine the effects on its behavior resulting from faults. This includes both faults that occur internally, and faults that are fed in as inputs.

9. A set of subsystems consisting of two or more devices, which may or may not be disjoint, are examined to determine the effects on their behavior resulting from internal and input faults.

10. The system as a discrete functional entity is examined to determine the effects on its behavior resulting from faults.

11. The spectrum of possible error behaviors, at the module, device, subsystem and system level, are grouped into classes that have substantially similar effects, called *eigenerrors*. These sets of eigenerrors will apply to all of, or some portions of, the system.

Error Injection Level (under construction)

12. The duty cycle of each functional module is parameterized (load-dependent)

13. The percent utilization and state of each functional module is parameterized (load-dependent)

14. The fraction of each of the various eigenerrors likely to result from the faults in various locations in each module, device and subsystem are modeled

15. The rate of each type of eigenerror is modeled. This result constitutes a high fidelity input to a Fault Injector

6. INITIAL PHYSICAL LEVEL FAULT MODEL

Space Environments

For this study we examined two mission orbit profiles relevant to the expected domain under which COTS parts might be used:

1. Geosychronous or deep-space applications, where the environment is dominated by galactic cosmic rays (GCR) and occasional solar flares. The solar flares contain high-energy protons as well as heavy charged particles, and occur at random times. The GCR flux for space missions near the earth or close to the sun is modulated by the solar cycle, and is about four times

lower during peak solar activity (solar maximum) compared to solar minimum conditions.

2. A low-inclination (≈ 28°) low-earth orbit (600 km) where the GCR flux is heavily shielded by the earth's magnetic field, and the only significant effect is from protons in the earth's Van Allen belts. Solar flares have little effect on the radiation level in this orbit. The proton flux increases when the spacecraft goes through the South Atlantic Anomaly (SAA). Although high inclination (polar) orbits have not yet been considered, the polar portion of the flight is similar, in increased proton flux and concomitant SEU rate increase, to that of the SAA in a low inclination orbit.

GCR and proton spectra were determined for these orbits using the AP-8 and CREME96 models [Ref. 7], assuming an external 100-mil aluminum shield surrounds all of the electronics of interest. These spectra were used to estimate the error rate (orbit averaged for the LEO case), as described in the following subsection.

Semiconductor Technology Model

The baseline semiconductor technology used to estimate error rates was a 0.18 μm epitaxial CMOS process, which is the process that is currently being used to produce high-performance microprocessors. The power supply voltage is assumed to be 2 V. In the future we will extend the analysis to include more advanced technologies, including an SOI process with 0.12 μm and smaller feature sizes.

Although no radiation effects data exists (to our knowledge) for devices fabricated in this process, it is possible to use scaling algorithms from the semiconductor and space radiation effects communities to determine how advances in device technology will affect the error rate of registers and other internal storage elements. The unknown factor is how transients in high-speed internal logic, which is a large part of a microprocessor chip, will be affected by scaling. One would normally expect the upset sensitivity to increase as devices are scaled because the critical charge required to upset the cell decreases with scaling. However, the cell area also decreases, and the issue of how scaling affects upset is far more complicated than indicated by elementary scaling calculations based on constant voltage or field. The upset rate for the calculations in this paper are based on scaling algorithms of Reference 11, along with radiation test results for the PC603e processor [9]. The initial analyses assume that logic errors are unlikely to occur because nearly all of the logic is clocked.

The dependence of upset on linear energy transfer (LET) was determined by first assuming a threshold LET of 0.02 pC/μm. Commercial microprocessors have had threshold LETs near that value over approximately a ten-year time period, even though feature sizes have decreased by more than a factor of ten as devices evolved [11]. The cross section was assumed to rise sharply to a saturation value that is related to the drain area, but allows for lateral charge collection from ions that strike near the sensitive drain region. The error rate was determined by integrating the GCR spectrum with the linear-energy transfer curve, allowing the angular dependence to extend to angle of 60°. The effective LET was assumed to scale with the secant of the incident angle out to that angle; for higher angles charge collection was assumed to be shared by several adjacent cells. A similar approach was used for protons, taking the proton spectrum and the spectrum of recoil products into account. The result of these calculations is an error rate for registers and other storage elements.

Multiple-bit faults were estimated by first considering experimental results for older memory technologies, which show that about 0.1% of the faults from high-energy particles produce multiple errors, and then estimating the multiple-bit rate for more advanced structures by taking the dimensions of the device structure into account. These calculations indicate that the number of multiple-bit errors will increase to 1-2% of the total number of faults for highly scaled devices, and that it is possible for 10 or more errors to occur for some geometrical paths [11]. While highly scaled devices are predicted to show this increase in faults, the exact nature of the increase or when it will begin to be manifested is uncertain and will be one of the research topics undertaken in the radiation testing and characterization portion of this project.

Circuit Design and Architecture

The error rate depends on software, circuit design and device architecture as well as the inherent sensitivity of individual storage elements. Clock tree faults were considered separately from register, memory and logic faults. Internal CPU L1 Cache fault rates were considered separately from register faults, because cache faults depend heavily on applications. They can also be tested independently of register faults.

The main uncertainty is that of estimating gate-level faults. Although the critical charge required to switch logic circuits is low enough to allow failures from heavy ions to occur, all logic is clocked, which makes the circuit sensitive to upsets only during the short period in which a clock transition occurs. Clocks in advanced microprocessors are very complex, and are designed to minimize skew and ensure low noise throughout the clock distribution network. Therefore it is reasonable to assume that no SEU errors will occur within the clock. The gate fault rate can be approximated by first considering the clock rate, and making some basic assumptions about the time interval in which latches or random logic will be sensitive to logic upset. At very high clock rates this approach may break down because the time interval over which charge from an SEU strike is collected may extend to about 0.5 ns, but it is probably a good assumption for clock speeds below 1 GHz.

Although registers are sensitive to upset at low LET, the cross section near threshold is also quite low. The cross section typically increases by three or more orders of magnitude until the LET is about 6 times greater than the threshold value. These results suggest that logic faults, which require a sustained single-level input signal or rapid transition, will have a significantly higher LET than register faults.

The gate-fault rate also depends on the fraction of gates in the logic tree that are sensitive to upsets (only some of the gates are used in ways that will affect the processor) as well as in the logic configuration and logic state. The net sensitivity of a processor to logic errors depends on three factors: (1) the logic transition time interval, (2) the LET threshold of the logic elements, and (3) the logic configuration and use. A conservative estimate of the first two factors reduce the upset rate of logic by a factor of about 10^{-3} compared to that of registers. The third factor is difficult to estimate, but probably reduces the overall rate by at least another factor of 100. The net result is that we estimate the gate fault rate is about 10^{-5} times that of the register fault rate.

7. Initial Design Level Fault Model

The REE First Generation Testbed (FGT) is a 20 node, 40 processor multicomputer. It is being designed and manufactured by Sanders, a Lockheed Martin Company, and is a homogeneous instantiation of the Air Force Research Lab's ISAC (Improved Space Architecture Concept) Program architecture. The principle concern, from a fault tolerance as well as operational perspective is to understand and limit the potential for fault propagation through the system. A distributed-memory architecture (as opposed to a shared memory architecture) was chosen for this first instantiation of an REE system. In future testbeds, we may reexamine shared memory architectures and other design choices.

The FGT is a homogeneous system comprising 20 identical nodes. Each node consists of:

- 2 Power PC 750 Processors, each with 1MB of L2 Cache
- 128 MB of sharable main memory
- 1 PCI bridge
- 1 Node Controller comprising:
  - 1 StrongARM SA-110 Processor
  - 4 MB memory
  - 2 Myrinet interfaces providing 1.2Gb/s of bi-directional I/O
  - 1 8-port Myrinet switch
  - Boot Flash holding a Lynx real-time OS kernel and boot code

The node controller provides communication and overall node control for the node in this two level architecture. All communication is via the Myrinet network fabric.

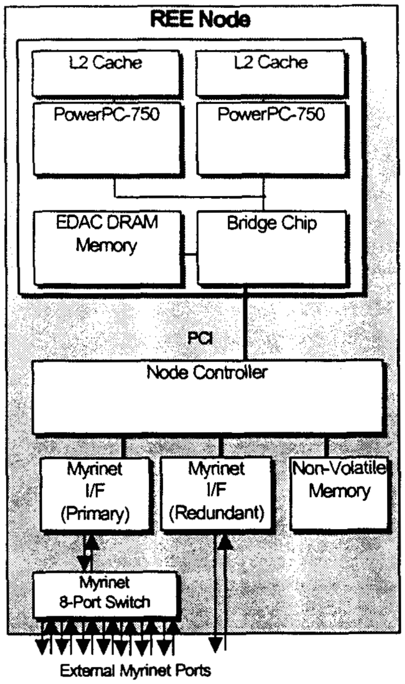

Figure 3: REE Node

A flight system, which is expected to be prototyped starting in 2002, will utilize then state-of-the-art COTS technology and may incorporate system-on-a-chip features, single-chip memory systems and any other advances made between now and then. The design level model for this implementation will need to be adapted for the CPU, interconnect mesh, and node/system architectures chosen for the flight system.

8. Current Results

We have modeled the projected fault rates using the REE First Generation Testbed (FGT) as a test case architecture for the two mission orbit profiles discussed earlier, under nominal and solar flare conditions.

Table 1 below shows the estimated register fault rate for the geosychronous/deep space orbit and low-inclination LEO orbit that was discussed earlier. These estimates apply to the baseline 0.18 μm epitaxial CMOS process, which is assumed to be the dominant process used in the components from which the FGT is fabricated. The "design-case" solar flare is taken from internal work done by JPL in 1992 that makes a more realistic assumption than the assumption of a worst-case flare, which is extremely unlikely and severely overestimates the effects of solar flares. An external 100-mil aluminum shield is assumed, which has little effect on the GCR flux, but affects the solar flare component.

Table 1. Calculated Error Rates for Two Mission Scenarios

Mission	Flare Condition	Faults per Bit-Day
Deep Space / GEO	none	3 x 10^{-5}
	design case	3 x 10^{-2}
LEO (28 °, 600 km)	none	2 x 10^{-4}
	design case	2 x 10^{-4}

Using these bit error rates, estimated fault rates are obtained by using the design level models. Several tables of results follow. Table 1 shows estimated fault rates for the FGT in low-inclination LEO. Note that the rates are the same for both nominal and solar flare activity, due to the Van Allen Belts shielding. Table 2 shows estimated fault rate for the FGT in the deep space/GEO environment under nominal solar activity. Table 3 projects the fault rate in that same environment during the occurrence of a design case solar flare. Each table breaks out faults by major components of a system node. Note that our model predicts that the projected gate faults rates in all cases are irrelevant compared to the latch fault rates for this technology. Note also that our model predicts that faults originating from the CPUs will be dominant in terms of number compared to all other components, by a factor of 5. Table 5 collects the CPU and node controller (NC) fault data into a single place for comparison.

If these model estimates turn out to be correct, operation in deep space or GEO will be problematic during solar flares. This fault rate turns out to be approximately 17 per minute per CPU. This rate will be mitigated somewhat by the fact that faults at many points in the system, such as the CPU general registers, have a high likelihood (~50%) of being overwritten before affecting the system behavior. But the error rate resulting from this fault rate may still turn out to be too high to handle by normal software implemented fault tolerance techniques. Since shielding is effect against solar flare, it may be necessary to increase the shielding for mission which need to operate though these kinds of events. In other cases, it may be necessary to shut down the system during the event.

On the other hand, the fault rates for nominal solar conditions, and for LEO are predicted to be extremely benign. Preliminary experiments with a prototype SIFT system and candidate science application have been carried out at fault rates much higher than these. In fact, we have done experiments with a simulated fault rate as high as 2 per minute on a scene classification application, with forward progress still being made.

As mentioned earlier, we have assumed in this model that the main memories and L2 caches are SECDED protected. So the faults reported for those components in Tables 2 – 4 are for double bit errors under conditions of slow scrubbing (approx every 20 minutes). Table 6 gives the predicted single bit fault rates for those components which would be caught and corrected by scrubbing. It is obvious that EDAC protection on the main memory and the L2 caches is essential in reducing the fault rate to a manageable level. In particular, an unprotected L2 cache would result in 10 times the faults of the CPU itself. particular, an unprotected L2 cache would result in 10 times the faults of the CPU itself.

Table 2: Fault Rate Estimates for FGT in LEO mission conditions

	Parameters	Latch Faults/day	Gate Faults/day	Total Faults/day
[illegible]		[illegible]	[illegible]	[illegible]
Number of nodes per system	20			
Additional system-level elements		7.7	0.00	7.7
Additional Network Switches per system	20	96	0.00	96
[illegible]		[illegible]	[illegible]	[illegible]
Number of CPU's per node	2	334	0.03	334
Size of RAM per CPU, Mbytes	64	1.1		1.1
Size of L2 cache per CPU, Mbytes	1	0.02		0.02
Node Controller (NC) CPU		47	0.00	47
Size of NC RAM, Mbytes	4	0.07		0.07
Bus controller (PCI)		3.9	0.00	3.9
No of Network Interface Units(NIU)	2	6.4	0.00	6.4
Number of Network Switches	1	4.8	0.00	4.8
Misc (watchdog, clock, PHRC)		0.29	0.00	0.29

Table 3: Fault Rate Estimates for FGT in deep space/GEO mission, under nominal solar conditions

	Parameters	Latch Faults/day	Gate Faults/day	Total Faults/day
System (20)		[illegible]	[illegible]	[illegible]
Number of nodes per system	20			
Additional system-level elements		1.2	0.00	1.2
Additional Network Switches per system	20	14	0.00	14
[illegible]		[illegible]	[illegible]	[illegible]
Number of CPU's per node	2	50	0.00	50
Size of RAM per CPU, Mbytes	64	.16		.16
Size of L2 cache per CPU, Mbytes	1	.00		.00
Node Controller (NC) CPU		7.1	0.00	7.1
Size of NC RAM, Mbytes	4	.01		.01
Bus controller (PCI)		.59	0.00	.59
No of Network Interface Units (NIU)	2	.97	0.00	.97
Number of Network Switches	1	.72	0.00	.72
Misc (watchdog, clock, PHRC)		.04	0.00	.04

Table 4: Fault Rate Estimates for FGT in deep space/GEO mission

	Parameters	Latch Faults/day	Gate Faults/day	Total Faults/day
[illegible]		[illegible]	[illegible]	[illegible]
Number of nodes per system	20			
Additional system-level elements		1,152	0.03	1,152
Additional Network Switches per system	20	14,458	0.60	14,458
[illegible]		[illegible]	[illegible]	[illegible]
Number of CPU's per node	2	50,108	3.90	50,112
Size of RAM per CPU, Mbytes	64	161		161
Size of L2 cache per CPU, Mbytes	1	2.5		2.5
Node Controller (NC) CPU		7,108	0.00	7,108
Size of NC RAM, Mbytes	4	10		10
Bus controller (PCI)		589	0.18	589
No of Network Interface Units (NIU)	2	965	0.60	966
Number of Network Switches	1	723	0.30	723
Misc (watchdog, clock, PHRC)		44	0.03	44

Fault Model Verification

To verify the Fault Models, the first step is to experimentally determine Latch Fault Rate and Gate Fault Rate parameters. It is possible to experimentally distinguish latch from gate faults by varying the clock and noting the difference in system fault rates/responses. If the chip supports it, some gate faults can be seen alone using a static clock. For verifying the predicted levels of faults in the individual functional modules making up each device, we are developing software-implemented fault detection and characterization techniques which should be nearly adequate to this task, and we will design a diagnostic FPGA for hardware-based fault characterization for the cases software cannot cover.

Table 5: Faults per Day for the three mission orbit profiles, broken out per CPU and per Node Controller (NC)

	LEO		GEO/DS	
	CPU	**NC**	**CPU**	**NC**
Nominal	167	47	25	7
Solar Flare	167	47	25,000	7,100

Table 6: Single bit fault rates (faults per day) for L2 caches and main memory

Environment	1 Mbyte L2 Cache	64 Mbyte Main Memory RAM
LEO	1,700	107,000
DS/GEO Nominal	250	16,000
DS/GEO Flare	252,000	16,106,000

Verifying the higher level error models is more straightforward, and will be done by running synthetic OS and applications tasks under irradiated conditions, and monitoring nominal and specially instrumented software outputs.

9. Status and Preliminary Conclusions

As a first-order test of the feasibility of the REE goal of using SIFT to mitigate radiation faults in space, prototype versions of the REE SIFT system software facilities have been demonstrated on a simplified multiprocessor system of 4 to 8 CPU's, with a variety of parallel processing applications. The majority of the fault injection experimentation work, to date, has been performed using a texture analysis image processing application which will potentially be used for autonomous navigation and geology by a future Mars Rover. This work was not performed by the authors of this paper, and will be presented in other publications. Here, we briefly summarized the results to date:

Experimental Results

For this first test, the operating system, application initialization, and initial data loading into the application was ignored. Injection was preformed on application data processing, and application data output I/O. Under these conditions, the prototype SIFT system has an overhead of <10% in the absence of faults. This simplification is reasonable for an initial rough estimate due to the nature of these types of jobs, i.e., tasks in which the science application will be consuming the vast majority of the computational resources, and will thus have the largest fault cross section. It should be noted that the high-rate/volume I/O in these tests is considered to be part of the application rather than the OS. These I/O operations will consume a significant fraction of the system resources in typical space based science applications. With these caveats in mind, we have found that with early prototype SIFT facilities, effective science computations have been successfully carried out with faults being injected into a science application at rates of one fault per minute. Naturally, a performance penalty is paid, but it is still only a maximum of 15% under these conditions which, per our current fault model, is a realistic estimate of the expected fault rate in a LEO or GEO orbit.

Radiation test results on older processors have shown that use conditions have a pronounced effect on the overall processor error rate [9,12]. Register-intensive tests that sequence through all registers produce a much higher cross section -- approximately three orders of magnitude greater -- than tests using an operational program such as a fast-Fourier transform or sort program. This is simply a result of the fact that many of the registers are either not used, or are rewritten before errors can propagate to the point that they affect the results. The error rate increases significantly when internal cache memories are used, approximately scaling with memory size.

Some errors result in conditions where the processor cannot be reinitialized without rebooting, or in some cases by temporarily removing power [9,12]. This class of error is obviously of critical importance in system applications, and may depend on the operating system as well as on the internal conditions in the processor. Fortunately, the error rate for such conditions is at least three orders of magnitude below the error rate for register errors. This suggests that only a very small region within the processor is responsible for such errors. However, it is difficult to characterize such errors, and more effort needs to be spent in determining how they relate to processor architecture.

Faults in caches and Memory Management Units (MMUs) have been a point of major concern for the REE system, because they are potentially very disruptive. Intuitively, this is because of the possibility that, for example, an error in one or more address bits in a Translation Lookaside Buffer (TLB) might cause multiple data values in an entire cache line or MMU block to be read from or written to the wrong location. Little work has been found in the literature on detailed analysis of the effects of faults in caches or MMUs, or in adding fault tolerance to them. The work that has been done is mostly focused on hardware-level voting, using replicated hardware synchronized at clock rates [7], and is too power-intensive for REE.

A detailed investigation of the MMU and cache subsystems of the PowerPC 750 has been performed, and the results are summarized below. A primary factor affecting the likelihood of errors from low-level faults is the time that elapses between the data being refreshed with new values. If values are rapidly being refreshed, faults have less time to propagate into errors, and fault latency is lower. As an example, faults in CPU general registers are typically overwritten about 50% of the time before causing any errors whatsoever. Depending on how they are used by the particular OS, there are a few hardware elements that may be used with either very short refresh latencies, or very long latencies. These include the 8 Block Address Translation (BAT) caches, and the 16 Virtual Segment Identifier (VSID) registers. By using suitable software structures, we believe it is feasible to use the other MMU and cache hardware elements so they are intentionally flushed and refreshed on a frequent basis. This occasional flushing will impose only a relatively small performance impact on operation, and will reduce the degree of error propagation from each fault. It will not prevent errors from occurring, however, and those errors that do occur, especially in the Translation Lookaside Buffers (TLBs) can cause *aliasing* of memory addresses. In this effect, faults in the address cause errors only under two circumstances.

(1) the faulty cache line was marked as dirty, and would be written back to memory

(2) executing code or data references the faulty address before it is flushed from the cache or MMU.

In either case, one or more incorrect data values will be introduced into the computation for as long as the aliased entry exists in the TLB.

To determine the likelihood of MMU- and cache-induced errors, the following table shows the error distribution by hardware for the node CPUs, MMUs, L1 caches and L2 caches, in the GEO environment with nominal solar activity analyzed above.

We see that over 1/3 of the faults per day will be in the TLBs, and roughly 1/2 will be in data stored in the L1 cache, while the remainder of all other CPU faults will be only a small fraction of the daily faults. The tag bits of the caches and MMUs have an extremely small fault cross-section, and so entries incorrectly marked as dirty can be ignored. For case (1) above, the window of opportunity for an error to occur due to a fault is very short, lasting from the time the data is changed until it is written back to memory. In write-back mode, this window could be a substantial number of clock cycles, but in write-through mode it would be effectively zero. Thus, by paying some performance penalty and using caches in write-through mode, this source of faults can be significantly reduced. For case (2), a critical question to determine is the likelihood of address aliasing, which will need to be determined by examination of memory organization and usage patterns for the OS and applications. This work is underway, and will be reported on in the future.

Table 7: Node CPU and L2 cache fault rates for GEO environment and nominal solar activity

Hardware Element	**Estimated Faults/day**
CPU Total:	25
L1 cache TLBs	2.2
L1 cache data	13
MMU TLBs	0.5
L2 cache TLBs	8.4
L2 cache data	0 (EDAC)
Other	1.2

Preliminary Conclusions

Our preliminary conclusions concerning some aspects of the REE system design are based on partially completed functional level and error level models, and include:

- During nominal solar activity, and in LEO orbit, the predicted fault rates are relatively benign. We expect to be able to adequately detect and handle these fault using only software fault tolerance techniques.
- Under solar flare conditions, operation of a COTS based system like the REE First Generation Testbed will be challenging. Heavy shielding may be able to bring the fault rate down to a manageable level.
- CPU faults are overwhelming concentrated in the L1 cache and TLBs. Scrubbing techniques may prove useful in mitigating the effect of these faults.
- Adding EDAC to the external L2 cache is essential because of the need to correct single-bit errors, and perhaps, in future technologies, multi-bit errors for highly scaled devices. This will raise cache fault tolerance to acceptable levels.

10. ACKNOWLEDGEMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

11. REFERENCES

[1] Tadashi Takano, Takahiro Yamada, et al, "Fault-Tolerance Experiments of the HITEN Onboard Space Computer," Proceedings of the Annual International Symposium on Fault-Tolerant Computing, IEEE, 1991, Austin, Texas.

[2] Bill Stapor, Pat McDonald, "Clementine 2 PC603e Radiation Effects Study," Technical Report from Innovative Concepts, 8200 Greensboro Drive, Suite 801, McLean Virginia 22102, (703) 893-2002, November 25, 1997.

[3] A. H. Johnston, "Scaling Effects and Radiation Susceptibility of Microprocessors," Technical Memorandum, Jet Propulsion Laboratory.

[4] Peter Liden, Peter Dahlgren, Rolf Johansson, Johan Karlsson, "On Latching Probability of Particle Induced Transients in Combinatorial Networks," Proceedings of the Annual International Symposium on Fault-Tolerant Computing, IEEE, 1994, Austin, Texas.

[5] Marcus Rimen, Joakim Ohlsson, Jan Torin, "On Microprocessor Error Behavior Modeling," Proceedings of the Annual International Symposium on Fault-Tolerant Computing, IEEE, 1994, Austin, Texas.

[6] Joakim Ohlsson, Marcus Rimen, Ulf Gunneflo, "A Study of the Effects of Transient Fault Injection into a 32-bit RISC with Built-in Watchdog," Proceedings of the Annual International Symposium on Fault-Tolerant Computing, IEEE, 1992, Boston, Massachusetts.

[7] Chung-Ho Chen, Arun K. Somani, "A Cache Protocol for Error Detection and Recovery in Fault-Tolerant Computing Systems," Proceedings of the Annual International Symposium on Fault-Tolerant Computing, IEEE, 1994, Austin, Texas.

[8] K. W. Li, J. R. Armstrong, J. G. Tront, "An HDL Simulation of the Effects of Single Event Upsets on Microprocessor Program Flow," *IEEE Transactions on Nuclear Science,* Vol. NS-31, No. 6, Dec. 1984, pp. 1139-144.

[9] F. Bezzera, et al., "SEE Test of Commercial Off-the-Shelf Microprocessors," Data Workshop from the 1997 RADECS Conference, p. 41, Cannes, France, September, 1997.

[10] J. L. Barth, *Modeling Space Radiation Environments*, Section 1 of the Short Course presented at the IEEE Nuclear and Space Radiation Effects Conference, Snowmass, Colorado, July 21, 1997.

[11] A. H. Johnston, "Radiation Effects in Advanced Microelectronic Technologies," IEEE Trans. Nucl. Sci., 45, p. 1339-1354, 1998.

[12] C. K. Kouba and G. Choi, "Single-Event Upset Characterization of the 486-DX Microprocessor," 1997 IEEE Radiation Effects Data Workshop, p. 48, IEEE Doc. 97TH8293.